Segmentation of RGB-D indoor scenes by stacking random forests and conditional random fields

Thøgersen, Mikkel; Guerrero, Sergio Escalera; Gonzàlez, Jordi; Moeslund, Thomas B.

Published in:
Pattern Recognition Letters

DOI (link to publication from Publisher):
10.1016/j.patrec.2016.06.024

Publication date:
2016

Document Version
Accepted author manuscript, peer reviewed version

Pattern Recognition Letters
journal homepage: www.elsevier.com

Segmentation of RGB-D indoor scenes by stacking random forests and conditional random fields

Mikkel Thøgersen[a,b,**], Sergio Escalera[b,c], Jordi Gonzàlez[b,d], Thomas B. Moeslund[a]

[a]*Aalborg University, Aalborg, Denmark*
[b]*Computer Vision Center, Barcelona, Catalonia Spain*
[c]*University of Barcelona, Catalonia Spain*
[d]*Universitat Autònoma de Barcelona, Catalonia Spain*

ABSTRACT

Depth images have granted new possibilities to computer vision researchers across the field. A prominent task is scene understanding and segmentation on which the present work is concerned. in this paper, we present a procedure combining well known methods in a unified learning framework based on stacked classifiers; the benefits are two fold: on one hand, the system scales well to consider different types of complex features and, on the other hand, the use of stacked classifiers makes the performance of the proposed technique more accurate. The proposed method consists of a random forest using random offset features in combination with a conditional random field (CRF) acting on a simple linear iterative clustering (SLIC) superpixel segmentation. The predictions of the CRF are filtered spatially by a multi-scale decomposition before merging it with the original feature set and applying a stacked random forest which gives the final predictions. The model is tested on the renown NYU-v2 dataset and the recently available SUNRGBD dataset. The approach shows that simple multi-modal features with the power of using multi-class multi-scale stacked sequential learners (MMSSL) can achieve slight better performance than state of the art methods on the same dataset. By using MMSSL, the method shows good improvements on the major performance metrics compared to the base models and this displays that the method is effective in this problem domain.

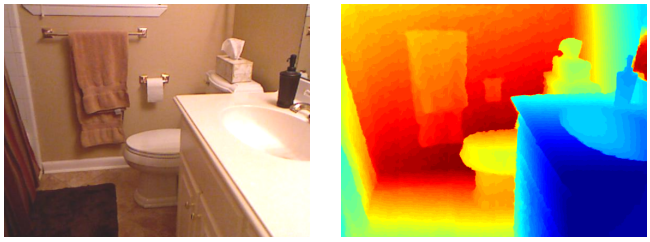
1. Introduction

Through cheap and readily available multimodal sensors, current and future robotics can gain a perception of their surroundings. To be useful, however, meaning has to be extracted from the sensor data. As such, semantic segmentation of scenes is a foundation for robots and machines to navigate and act in the human-centric world.

A number of works have investigated the problem of indoor semantic segmentation previously. The majority of them are focused on the NYU-v2 dataset provided in (Silberman et al., 2012). A sample image from the dataset is shown in figure 1. While the dataset has 894 classes in total, most works consider a semantic 4-class problem originally stated in (Silberman et al., 2012), others increase the number of classes to 13 and 40. In this work, 4-class segmentation is investigated.

The NYU-v2 dataset is a sequel to their first dataset, NYU-v1 (Silberman and Fergus, 2011), where the authors propose the initial solution to this problem and provide the first densely labeled RGB-D dataset for indoor semantic segmentation. Their approach is based on an initial segmentation followed by an evaluation of three types of potentials in an energy function: a unary appearance potential incorporating a range of feature descriptors including location priors, SIFT features etc.; a class transition potential and lastly a spatial smoothness term. In a more recent work, by (Khan et al., 2014), the dominant lines and vanishing points of the scene are used to align the scene to the major surfaces, e.g. the floors and walls. Using a combination of color and depth based edges the scene is superpixelated using a k-means clustering, the resulting regions are subsequently fitted with a plane and features are extracted and combined in a Conditional Random Field (CRF).

**Corresponding author: Tel.: +45-26714763;
e-mail: mt@hst.aau.com (Mikkel Thøgersen)



(a) (b)

Fig. 1. (a) A sample scene from the NYU-v2 dataset; (b) Normalized inpainted depth map of the same scene.

In (Müller and Behnke, 2014) they used a Random Forest (RF) to do an initial pixel-wise class prediction, as originally proposed in (Stückler et al., 2013). This initial segmentation is later aggregated in a Simple Linear Iterative Clustering (SLIC) segmentation on which a CRF infer the final class labels. Their approach showed state-of-the-art results. A multi-scale CRF is proposed in the more recent work (Hamedani and Harati, 2014). Here a downsampled version of the image is inferred and the predictions are propagated through a multi-scale CRF pyramid. They also explore the use of a temporal pairwise potential to enforce the beliefs through video sequences.

In (Puertas et al., 2015) they present the Multi-class Multi-scale Stacked Sequential Learning (MMSSL) framework. The idea is to stack subsequent classifiers and cumulatively extend the feature sets with filtered versions of each classifiers predictions. Results show improved performance in 1D and 2D sequential problems.

Likewise, (Cohen and Carvalho, 2005) shows how using a base classifier with a consecutive classifier acting on the predictions of the base classifier can improve segmentation results considerably in 1D sequential problems. To obtain the improved performance, a contextual feature is created from the confidence map of the base classifier, and this is where the method gains its discriminative power, especially when combined with non-contextual classifiers. They explore a number of different classifiers and evaluate stacked versions. Their results on some datasets are impressive, especially they test a stacked CRF and find that it can improve the results of the standard CRF. Their work is further developed in (Gatta et al., 2011) where they add a multi-scale decomposition acting as the contextual feature. Instead of a single distance of contextual information, as used in (Cohen and Carvalho, 2005), multiple distances are used which outperforms previous methods.

Also in (Sampedro et al., 2014), classifiers are stacked iteratively until some stopping criteria. They call it the Iterative Multi-class Multi-scale Stacked Sequential Learning or IMMSSL. The paradigm is used on 3D medical volume scans, but has not previously been tested on 3D data for indoor semantic segmentation which is not sequential in the same manner as for volume imaging.

In this work we explore and enhance current work on semantic segmentation in indoor cluttered scenes. Specifically, this paper focuses on introducing the MMSSL framework on the methods that have previously been proven effective for the semantic segmentation problem (Müller and Behnke, 2014; Hamedani and Harati, 2014; Khan et al., 2014). The resulting model is trained and tested on the NYU-v2 dataset and the SUNRGBD dataset (Song et al., 2015) which contains reformatted data from the NYU-v2, Berkeley B3DO (Janoch et al., 2013) and SUN3D (Xiao et al., 2013) datasets. The data in SUNRGBD is captured from four different RGBD sensors and contains eight times as much data as the NYU-v2 dataset. This works shows the first four-class segmentation attempt of the SUNRGBD dataset and compares this to the original four-class problem posed with the popular NYU-v2 dataset.

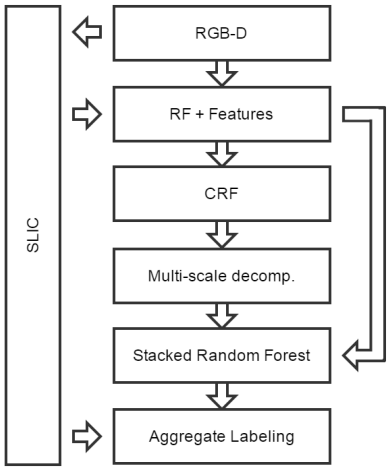


Fig. 2. Model overview. Features are extracted from the RGB-D images, aggregated in superpixels and fed to the CRF. A multi-scale decomposition of the predictions from the CRF is created and used in an RF which predicts the final labels.

The rest of the paper is organized as follows: Section 2 presents the approach for semantic RGB-D scene segmentation, describing features and stacked classifiers. Section 3 shows the experiments. Finally, Section 4 concludes the paper.

2. RGB-D Scene Segmentation

Our proposed technique for RGB-D scene segmentation is a combination of the methods of previous works with the addition of the MMSSL framework. Several of the related works, and in particular the state of the art, use the SLIC segmentation. Because of its proven performance and efficiency it is also adopted as the main segmentation in this work. An initial pixel-wise RF, similar to the one presented in (Müller and Behnke, 2014), is used to give an initial approximate class label. It is based on random offset features, which have a strong resemblance to HAAR features, as used in (Viola and Jones, 2004). The implementation is not the exact implementation presented in (Müller and Behnke, 2014), but with similar characteristics. The confidence map from the RF classifier is used as a feature amongst an array of other features as the input to a CRF. This constellation with an RF and a CRF makes up the base classifier of the MMSSL construction.

Applying the base model onto the training data gives class confidences for each pixel. The idea of MMSSL is to use these confidences of the base classifier as a super feature and

Table 1. Parameters for the initial Random Forest.

Parameter	value
Num. of features	40
Num. of samples	$1.56 \cdot 10^6$
Num. of trees	20
Num. of sample at each decision split	7
Minimum num. of samples at leaf	10

combine it together with the original features as the input to a stacked classifier. The stacked classifier work consecutively and can actively learn the mistakes of the base classifier, correct them and improve the overall performance. As mentioned before (Gatta et al., 2011) used a multi-scale decomposition of the confidences of the base classifier instead of just using the confidences. This improved results and a similar feature is constructed here.

This super feature combined with the original feature set is used as the input for another RF, which acts as the stacked classifier. Finally the classifications of the stacked RF are aggregated back into the original SLIC segmentation to make the classifications more coherent. The complete model pipeline is depicted in figure 2.

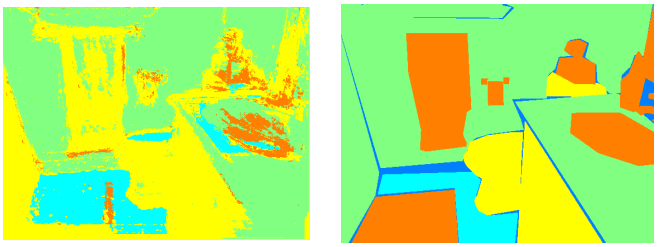
2.1. Random Forest

An initial estimate of the classes is obtained through the use of a Random Forest with random offset features that captures subtle details of either the depth channel or the Lab color space channels. This approach is similar to the one originally proposed in (Stückler et al., 2013). For every pixel, two random offsets within a certain range are chosen. A randomly sized rectangular patch around each offset are summed; the feature now consists of the absolute difference between these two sums. Obviously this has similarities to the Haar-like features, (Viola and Jones, 2004) and enjoys the same benefit of integral images.

Initially thousands of randomized offsets and patch sizes are trained on the data. Following, the out-of-bag samples are tested with the trained model and by using the errors of these, the discriminative power of each of the random features can be evaluated. Based on this evaluation, by looking at the out of bag error, a subset of the random features can be selected and the model is retrained to obtain the first RF model. The parameters of the RF are shown in table 1 and the corresponding response from the trained RF is shown in figure 3. Finally, the predictions of this classifier are aggregated in a segmentation and used as a feature in the CRF classifier.

2.2. Segmentation

Similar to other state of the art approaches (Müller and Behnke, 2014; Silberman and Fergus, 2011; Silberman et al., 2012) the proposed system is centered around a CRF that acts on an over segmentation of the input imagery. The outcome is a classification of the individual superpixels. To obtain the super pixel segmentation, the localized k-means based SLIC algorithm (Achanta et al., 2012) is used in this work, which is renown for its speed and ability to create near uniform segments while preserving contours. It is an important part of the model, as it defines the boundaries of the segmentation and some features are derived from it.



(a) (b)

Fig. 3. Shows the Random Forest response together with the ground truth on the right.

2.3. Generic Features

A set of primitives are extracted from the data, as these are used as a basis for several of the features. These are the trivial Cartesian coordinates and the normals. The normals are calculated as the cross product estimates (Klasing et al., 2009). Several more elaborate methods are available, however in this context the normals will be averaged over regions which will act as a filter.

From the obtained data, a set of features are readily available. For the individual superpixels, the features include: Color using the Lab color space, the normal and the standard deviation of the normal which captures curvature in either direction. Also, a blurred gradient of the image and the depth map are included to provide clues on areas with high change of color and depth. For the case of pairwise potentials, all of the above and 3D Cartesian coordinates are used. Most are calculated as the absolute difference with the exception of the color and normal features. The pairwise color feature is modulated to penalize in a non-linear fashion:

$$f_{\text{colDiff}}(R_1, R_2) = \exp(-\beta \, \|\mathbf{c}_1 - \mathbf{c}_2\|^2), \tag{1}$$

where R_1 and R_2 are two adjacent superpixels, $\mathbf{c}_1$ and $\mathbf{c}_2$ are the Lab colors of the superpixels and β controls the attenuation of the feature. The normals are compared using a normalization of the difference of angles and they are also compared w.r.t their inclination in spherical coordinates.

2.4. Dominant normal direction

An important concept of the model is estimating the layout of the room and finding the dominant axes of the room, e.g. the floor and wall normals. The Manhattan principle is the assumption that lines and edges in indoor imagery are aligned with either the floor or the walls (Coughlan and Yuille, 2003). Extending this to surfaces is a natural step, as the edges and lines only exist where surfaces meet. Based on this extended Manhattan principle - that major surfaces are aligned with the floor or the walls - the normals of the scene are investigated using a mean shift clustering, usually providing 4 to 8 candidate dominant directions. To illustrate these dominant directions, a visualization of the normals and their distribution for a given scene is shown in figure 4. As several authors (Müller and Behnke, 2014; Silberman et al., 2012; Gupta et al., 2013) similarly have noticed, the floor normal can be found by taking the candidate nearest

(a) (b)

Fig. 4. (a) Shows a scene plotted in 3D Cartesian coordinates, besides it, in (b) a plot of the concentration of normals is shown, it is created using a histogram of the normals, where the concentrations are plotted as elevations on a sphere, corresponding to the direction of the normals. Notice the three main bulges (red color), they correspond to the three dominant normal directions in the image.

to the upwards axis of the estimated Cartesian coordinate system. While the previous works use methods like clustering the normals into 10 clusters and choosing the one which is closest to alignment with the y-axis, the method used here is somewhat more elaborate and can also find walls. In this model, the candidates, $\boldsymbol{P}$, from the mean shift clustering are evaluated using the following expression:

$$\mathbf{n}_{\text{floor}} = \underset{p \in \boldsymbol{P}}{\text{argmax}} \ \exp\left[-\left(\frac{|p_\theta - \theta_{\text{std}}|}{180} \right)^{\lambda_{\text{f}}} \quad \left(1 - \frac{p_\mu}{\sum\limits_{p \in \boldsymbol{P}} p_\mu} \right) \right]. \tag{2}$$

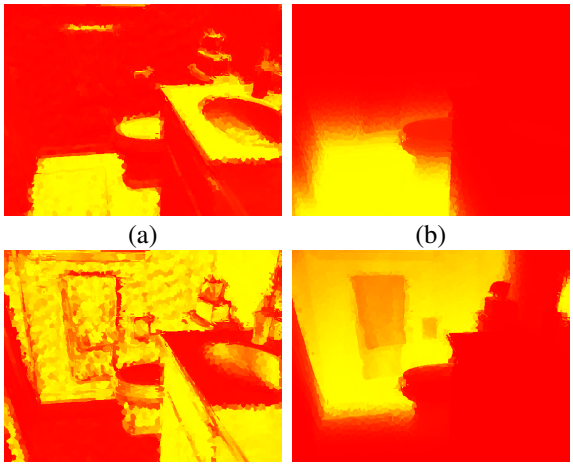
In addition to choosing the candidate, p, with a normal direction, p_θ, close to some standard floor inclination, θ_{std}, the evaluation ensures that there is sufficient support - e.g. the number of normals, p_μ, pointing in the same direction as the candidate. Also, the importance of inclination compared to support is controlled by adjusting the λ_{f}-parameter.

Once the floor direction is determined, the wall normals are found in a similar fashion from the remaining dominant normal candidates. When finding the wall normals, the assumption is that candidate directions are perpendicular to the floor normal and to other candidates - assuming that rooms are rectangular. This does not hold for rooms with more complex walls, but performs well in practice. Initially the vertical angle differences, $p_{\hat{\theta}}$, between the floor normal and the candidates are calculated and 90° is subtracted, so that candidates with an angle difference close to zero are more likely to be wall. Afterwards the process continues iteratively following these steps:

- (This step is skipped on first iteration). The candidates are compared with the found wall normals in the horizontal direction by finding the angle difference, $p_{\hat{\varphi}}$, and subtracting whichever multiple of 90° is closest to the found value. This has the effect that candidates that are perpendicular or opposing to the already found wall normals are most likely to be wall.
- On every iteration the remaining candidates are evaluated using:

$$\mathbf{n}_{\text{wall}} = \underset{p \in \boldsymbol{P}}{\text{argmax}} \ \exp\left[-\left(\frac{|p_{\hat{\theta}} + p_{\hat{\varphi}}|}{180} \right)^{\lambda_{\text{w}}} \quad \left(1 - \frac{p_\mu}{\sum\limits_{p \in \boldsymbol{P}} p_\mu} \right) \right], \tag{3}$$

where λ_{w} controls the importance of either term. If the evaluation equation returns a value greater than some threshold, the candidate is accepted as a wall normal, otherwise the process is stopped.



(a) (b) (c) (d)

Fig. 5. Normal features based on the dominant normal directions. The original image is shown in figure 1. (a) Intensity image of the vertical normal feature. (b) Intensity image of the floor probability. (c) Similar to (a), the wall normal feature compares the normals to the found wall normal directions. Lastly, (d) is similar to (b), using the normals of the walls.

Knowing the normals of each structure entity, their surfaces are found in the scene by fitting a plane to a point on the surface. Using the assumption about rectangular rooms, the furthest point in the opposite direction of one of the entities normals must be a point on the entity. Having a point and a normal for an entity, a plane can be constructed and used for assigning a wall and floor probability.

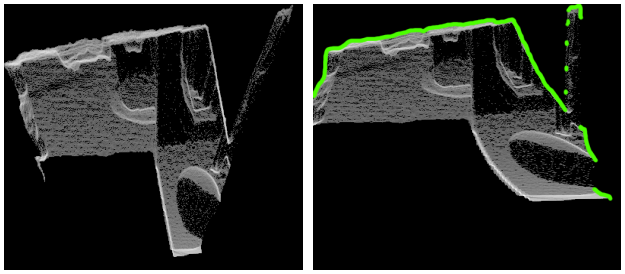
Because the floor normal is known, it is possible to find all upwards pointing surfaces by comparing normals using the scalar product and an exponentiation:

$$f_{\text{flat}}(\mathbf{n}) = (\mathbf{n} \cdot \mathbf{n}_{\text{floor}} - 1)^{\alpha}, \tag{4}$$

where α attenuates the feature. These normal features, amongst other similar features, are shown in figure 5.

The floor plane provides a means of finding the height of each individual superpixel. The height is divided into bins to better accommodate the nature of CRF features. The bins are set to span 2.5 meters, assuming that most scenes do not exceed this height, in case they do they are assigned to the last bin.

Another feature, derived from the dominant normal direction, is the room layout feature. The idea is simple: flipping the 3D scene, such that it is viewed from above. Once this is done, the walls will be the bounding periphery and can be detected by following the camera rays out in every direction, see figure 6. By looking at the camera rays and finding the last point registered in each line from the camera, the last point will approximately be a part of the wall. Following, a probability can be assigned to all points that are vertically aligned with this furthest point, effectively separating the walls and the rest of the scene. In this work, the feature is a complimentary feature to the dominant normal directions. This feature and assumptions are inspired



(a) (b)

Fig. 6. (a) Room layout view of the sample image shown in figure 1. The 3D point cloud has been rotated, such that a 'top view' of the scenery is obtained. (b) By warping the image to the field of view of the kinect camera this image is obtained. The walls can be found by following each vertical line from the bottom and up and stopping when there is no gray pixels left (marked with green in the image). See an example of the probability assignments from the room layout feature on figure 8(b).

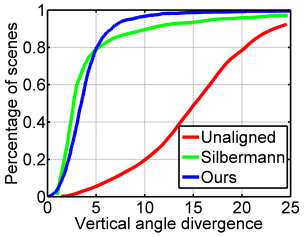
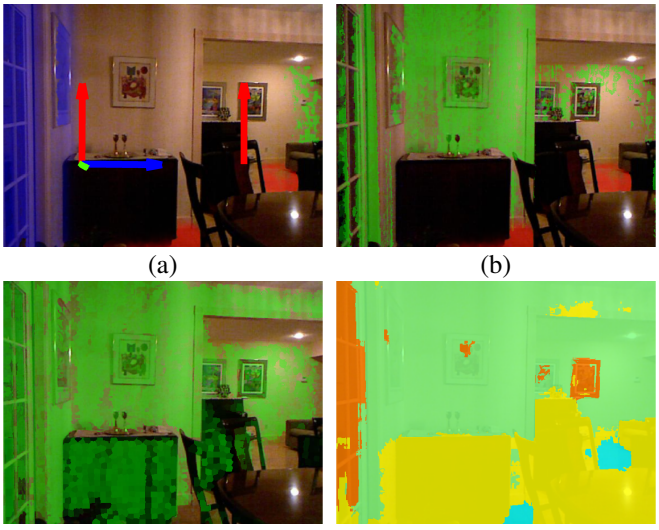


Fig. 7. Shows how well the found up-vector corresponds to the accelerometer data from the kinect (based on NYU-v2 data).



(a) (b) (c) (d)

Fig. 8. (a) Dominant normal directions and wall (blue + green) and floor (red) probability assignments. The arrows on the left show the dominant normal directions and the right arrow shows the accelerometer ground truth up vector. (b) Room layout feature. (c) Wall normal feature. (d) Final assignments.

by the work of (Cadena and Košecka, 2013). An example is given in figure 6.

To display the dominant normal directions features and how they interact with other simple features, figure 8 shows a scene with the corresponding features. The first image (a) shows the dominant normal directions as a coordinate frame in the scene. The red arrow to the right is the up vector measured by the accelerometer in the kinect camera when the image was taken. This can be considered as ground truth. Looking at the scene behind; the walls are assigned a probability by fitting a plane to them (the blue and green shade), however the door opening breaks the assumption of having purely rectangular rooms so most of the front wall is not registered (it should have a green color). Fortunately, the other features can redeem this. Image (b) shows the outcome of the room layout feature, it captures exactly the front most wall which the previous features did not. Finally, image (c) shows the wall normal comparison feature which generally includes any wall-aligned surface and therefore includes too much. The three features together captures different aspects and combined with the learners, the end result is shown in image (d).

To explore how well the scenes are aligned, the found up vectors are compared to that of the accelerometer data from the kinect. The comparison is shown in figure 7. It shows that the approach taken in this work is on par with (Silberman et al., 2012) and exceeds for getting most scenes within 10°.

2.5. Multi-scale Sequential Stacked Classifier

The MMSSL framework is a way to improve the results of a model by adding another discriminative classifier on top of the existing one. The stacked classifier is trained with a multi-scale decomposition of the confidence map of the base classifier as well as the original feature set. This effectively corrects errors of the base classifier.

The multi-scale decomposition is generated by analyzing the distribution of nearby pixels confidence maps in multiple distances from the point in question. For a point p with the pixels P_i situated in some half-closed distance intervals from p denoted by $i = \{1, 2, ..., n\}$, the multi-scale decomposition is the averaged distribution of the confidence map, C, of each class, $c = \{1, 2, ..., k\}$, in each interval. It can be described as:

$$a_{n(c-1)+i} = \frac{\sum_i C_c(P_i)}{|P_i|}, \tag{5}$$

where C_c refers to the confidence map for the class c. The vector a is the resulting feature vector for p. In effect, this decomposition creates a feature vector that is linear in size with the number of classes, *e.g* kn, as a consequence the framework is suited for problems with a relatively low number of classes, although it can be compressed when dealing with a large number of classes (Puertas et al., 2015). Notice that the points in P_i are found as the points with an Euclidean distance from the query point p falling in the interval i. A depiction of the decomposition is shown in figure 9. For each sphere in the image, the distribution of each class is found, these distributions then constitutes the super feature. In this work the distances used are: 2, 5, 11 and 30 cm.

This super feature or multi-scale decomposition is added to the initial feature set to form an extended feature set which is the input to the stacked classifier for the final classification. As a stacked classifier an RF is chosen which is trained on the extended dataset, and finally the predictions are aggregated in the superpixel segmentation giving the final labeling.

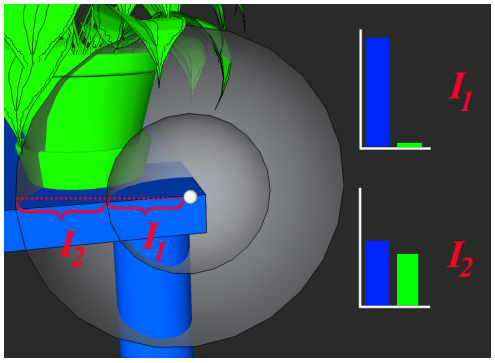


Fig. 9. Depiction of the multi-scale decomposition. From the query point in the center, there are two intervals, I_1 and I_2 that each generates a sphere. In each interval, the distribution of the confidence map is extracted, which gives the resulting feature vector.

3. Experiments

The proposed model is tested on the NYU-v2 and SUNRGBD datasets. The NYU-v2 dataset is composed of 1449 densely labeled RGB-D images captured using a Kinect camera. The dataset is provided with in-painted depth maps, solving the common problem of filling holes in data from depth-sensing technologies. The manual colorization method of Levin *et al.* (Levin et al., 2004) is used for the in-painting. The dataset includes a standard data split introduced with the first paper on the dataset (Silberman et al., 2012). This gives a total of 795 densely labeled training images and 654 test images spanning various different scenarios, from kitchens to public libraries. The SUNRGBD dataset consists of 10335 RGB-D images, 5284 training samples and 5051 test samples. The dataset comes with 37 annotated classes, however in this paper, only a four-class problem is considered and hence the 37 classes are translated into the four classes used here. Following the available literature on the subject, the two commonly compared measures are per class accuracy and pixel accuracy.

3.1. Ablation Study

To examine the importance of features and methods, different settings are tested. The baseline is given as the generic features with the conditional random field. Each setting is shown in table 3, where the feature groups have been abbreviated according to table 2. The CRF alone with the generic feature set is discriminative enough to get fair results on the three classes with least variance, whereas it fails to classify the props class. This is a natural cause of the fact that the props class spans a wide variety of objects making it hard to find discriminative features for such a class, particularly with only the generic feature set. Adding the normal features with especially the vertical normal feature gives better results on the props class, as it can be used to separate objects on tables and on the floor. In general, it increases the performance across all classes.

To test whether the room layout feature has a positive effect, it is tested by itself with the CRF, it adds some descriptive power however not as much as the normal features. This appeals to the intuitive sense, as it can only tell whether a given pixel is *inside* the bounding walls or is in fact the walls.

Table 2. Feature reference table.

Feature sets	Node	Edge
Generic features	G	
Lab color	•	•
3D position		•
Normal	•	•
Std. dev. of the normal	•	•
Blurred depth gradient magnitude	•	•
Blurred image gradient magnitude	•	•
Room Layout features	RL	
Room layout wall probability	•	•
Inverse room layout wall probability	•	•
Normal features	N	
Discrete height	•	•
Vertical normal comparison	•	•
Wall normals comparison	•	•
Floor probability	•	•
Wall probability	•	•
Continuous height		•

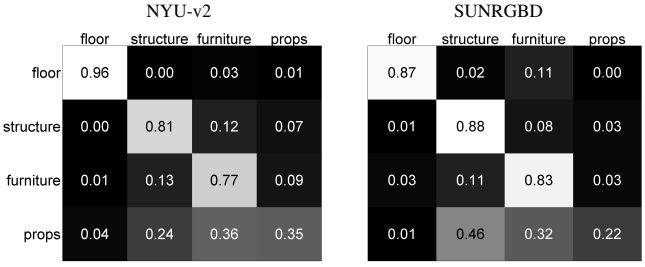


Fig. 10. Confusion matrix of the NYU-v2 and SUNRGBD results.

As expected, combining the two feature sets shows an increase on all measures. Adding the RF initial predictions gives a large boost to the props class. It captures subtle details of data set and this pays off. Unfortunately it takes its toll on the furniture class. This is probably due to the fact that the props label is often assigned to areas with a high change in contrast, which is also often the appearance of furniture. This effect is also visible in the results shown in figure 12. Finally adding the stacked RF and the multi-scale decomposition improves accuracy to all classes from the predictions of the base classifier. While the CRF is useful at pruning impurities in region classifications, it also has a tendency to merge across regions, where it is unwanted. The added contextual awareness of the multi-scale decomposition can prune these faulty labeled regions. The ablation study shows that the inclusion of MMSSL improves the performance on the major metrics of both datasets. One difference between the two datasets is the accuracy on the Props class; On the NYU-v2 this has good performance when the initial RF is introduced, however in the SUNRGBD this does not seem to be the case. This may simply be because of the difference in the datasets, but it could also be due to the translation from the 37 classes to 4 classes, where the Props class may be even more diverse than for the NYU-v2 dataset.

3.2. Results and Comparisons

In figure 12, sample results are displayed, along with the predictions at each step of the model. The initial segmentation using only the RF shows a very rough segmentation, with a high amount of impurities and scattered labellings. This is mended in the CRF with the SLIC segmentation, where the resulting la-

Table 3. Table of feature sets used in the ablation study. The abbreviations used in the table refer to the sets shown in table 2.

	Methods	Features	Floor	Struct.	Furn.	Props	Pr. class acc.	Pix. acc.
NYUv2	CRF	G	85.3	77.6	60.5	1.8	56.3	60.0
	CRF	G+N	92.8	77.3	78.0	14.5	65.7	68.9
	CRF	G+RL	83.0	76.4	74.2	2.7	59.1	64.1
	CRF	G+N+RL	94.3	78.8	**81.1**	13.8	67.0	70.6
	CRF+RF	G+N+RL	93.3	79.4	74.0	33.4	70.0	71.5
	CRF+RF+RFS	G+N+RL	**95.5**	**80.5**	77.1	**35.3**	**72.1**	**73.8**
SUN RGBD	CRF+RF	G+N+RL	85.8	**88.8**	**83.4**	4.8	65.7	79.8
	CRF+RF+RFS	G+N+RL	**86.7**	88.1	82.7	**21.6**	**69.8**	**80.9**

Table 4. State of the art comparison.

NYUv2	Floor	Structure	Furniture	Props	Per class acc.	Pix. acc.
Müller and Behnke (Müller and Behnke, 2014)	94.9	78.9	71.1	42.7	71.9	72.3
Couprie *et al.* (Couprie et al., 2013)	87.3	87.8	45.3	35.5	63.5	64.5
Khan *et al.* (Khan et al., 2014)	87.1	**88.2**	54.7	32.6	65.6	69.2
Gupta *et al.* (Gupta et al., 2013)	82	73	64	37	65	64.9
Nico Höft *et al.* (Höft et al., 2014)	77.9	65.4	55.9	**49.9**	62.0	61.1
Ours	**95.5**	80.5	**77.1**	35.3	**72.1**	**73.8**

SUNRGBD	Floor	Structure	Furniture	Props	Per class acc.	Pix. acc.
Ours	**86.7**	88.1	82.7	**21.6**	**69.8**	**80.9**

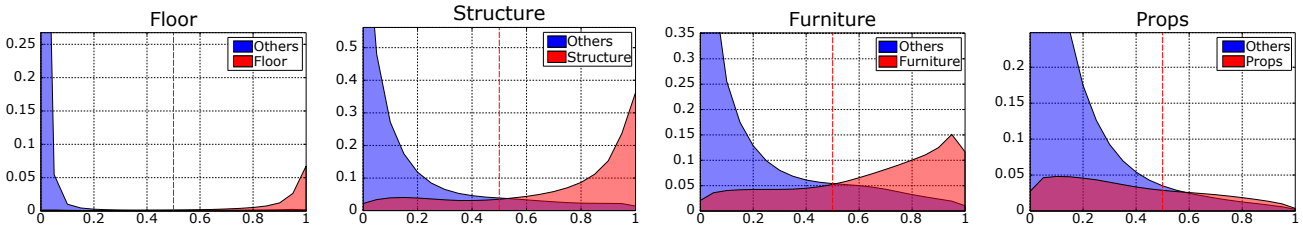


Fig. 11. Posterior class distributions for the NYU-v2 dataset. The x-axis shows the confidence level of the classified samples while the y-axis shows the normalized distribution of the samples. The red dashed line at 0.5 represents the classifier decision boundary.

bellings are coherent and respect most object boundaries. It has however a tendency to merge across boundaries. This problem is solved using the stacked RF; notice how in the third image column, that the hole underneath the arm rest of the chair is labeled correctly as floor. This is one of the ways the multi-scale decomposition works with contextual awareness.

To show the comparative performance, table 4 shows the results of current and previous state-of-the-art methods with the average class accuracy measure, which is equivalent to the mean of the confusion matrix diagonal and the pixel accuracy. The unlabeled pixels in the ground truth are disregarded for these measures. As shown in the table, our approach excel on some measures and obtains an increment in state-of-the-art results on the NYU-v2 dataset. The confusions matrix is also supplied in figure 10. Finally the posterior class distributions are shown in figure 11. The red area represents the class being investigated, while the blue area represents the other classes. It is clear from the distributions that the floor class is very separable, while the other classes are less separable with the features and classifiers chosen here. Worst is the props class, where most of the true samples are found to be beneath the decision boundary.

4. Conclusion and Discussion

We presented a model for semantic segmentation of cluttered indoor scenes. The method is based on the stacking of a Conditional Random Field and Random Forests. The results show that the model excels the CRF classifier and adds a heightened contextual awareness by including the multi-scale decomposition. The model is tested on two public datasets, the SUN-RGBD on which we show the first four-class performance results and the NYU-v2 where it shows comparable and better performance than state of the art.

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *Transactions on Pattern Analysis and Machine Intelligence*, 34:2274–2282.

Cadena, C. and Košecka, J. (2013). Semantic parsing for priming object detection in RGB-D scenes. *The International Journal of Robotics Research*, 34:582–597.

Cohen, W. W. and Carvalho, V. R. (2005). Stacked sequential learning. *Proceedings of IJCAI*, pages 671–676.

Coughlan, J. M. and Yuille, A. L. (2003). Manhattan world: Orientation and outlier detection by bayesian inference. *Neural Computation*, 15:1063–1088.

Couprie, C., Farabet, C., Najman, L., and LeCun, Y. (2013). Indoor semantic segmentation using depth information. *International Conference on Learning Representation*.

Gatta, C., Puertas, E., and Pujol, O. (2011). Multi-scale stacked sequential learning. *Pattern Recognition*, 44:2414–2426.

Gupta, S., Arbelaez, P., and Malik, J. (2013). Perceptual organization and recognition of indoor scenes from RGB-D images. In *Computer Vision and Pattern Recognition (CVPR)*, pages 564–571.

Hamedani, T. and Harati, A. (2014). Multi scale CRF based RGB-D image segmentation using inter frames potentials. *International Conference on Robotics and Mechatronics*, 2:920–925.

Höft, N., Schulz, H., and Behnke, S. (2014). Fast semantic segmentation of RGB-D scenes with GPU-accelerated deep neural networks. *Advances in Artificial Intelligence*, 8736:80–85.

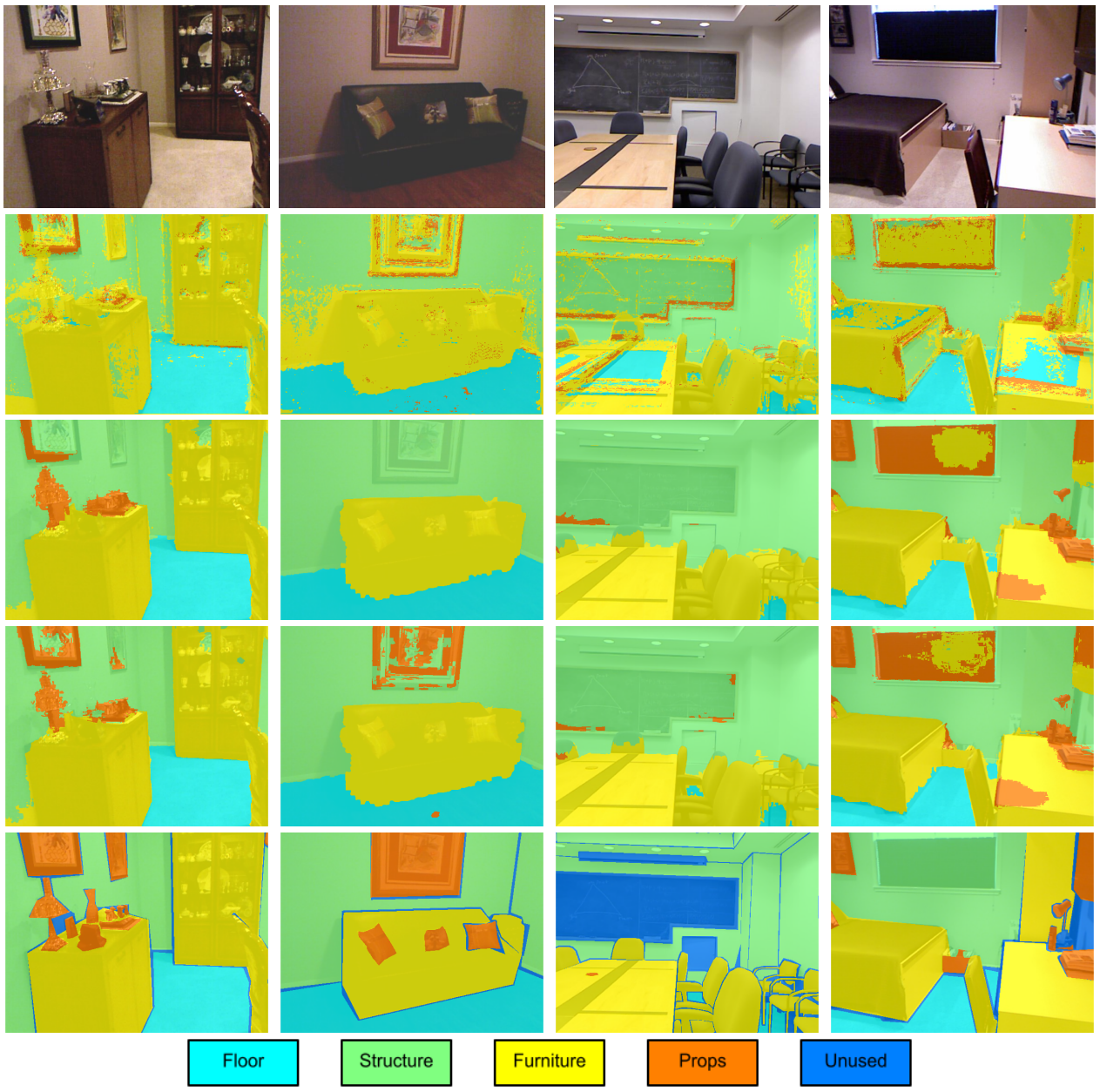


Fig. 12. Sample results. The rows corresponds to: 1) Original RGB images, 2) initial RF label predictions, 3) CRF predictions, 4) full model predictions, 5) ground truth and finally the class labels.

Janoch, A., Karayev, S., Jia, Y., Barron, J. T., Fritz, M., Saenko, K., and Darrell, T. (2013). A category-level 3d object dataset: Putting the kinect to work. In *Consumer Depth Cameras for Computer Vision*, pages 141–165.

Khan, S. H., Bennamoun, M., Sohel, F., and Togneri, R. (2014). Geometry driven semantic labeling of indoor scenes. In *European Conference on Computer Vision*, volume 8689 of *Lecture Notes in Computer Science*, pages 679–694. Springer International Publishing.

Klasing, K., Althoff, D., Wollherr, D., and Buss, M. (2009). Comparison of surface normal estimation methods for range sensing applications. In *International Conference on Robotics and Automation*, pages 3206–3211.

Levin, A., Lischinski, D., and Weiss, Y. (2004). Colorization using optimization. In *ACM SIGGRAPH*, SIGGRAPH, pages 689–694.

Müller, A. C. and Behnke, S. (2014). Learning depth-sensitive conditional random fields for semantic segmentation of RGB-D images. *International Conference on Robotics and Automation*.

Puertas, E., Escalera, S., and Pujol, O. (2015). Generalized multi-scale stacked sequential learning for multi-class classification. *Pattern Analysis and Applications*, 18(2):247–261.

Sampedro, F., Escalera, S., and Puig, A. (2014). Iterative multi-class multi-scale stacked sequential learning: Definition and application to medical volume imaging. *Pattern Recognition Letters*, 46:1–10.

Silberman, N. and Fergus, R. (2011). Indoor scene segmentation using a structured light sensor. *International Conference on Computer Vision*, pages 601 – 608.

Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. (2012). Indoor segmentation and support inference from RGB-D images. In *European Conference on Computer Vision*, pages 746–760.

Song, S., Lichtenberg, S. P., and Xiao, J. (2015). Sun rgb-d: A rgb-d scene understanding benchmark suite. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576.

Stückler, J., Waldvogel, B., Schulz, H., and Behnke, S. (2013). Dense real-time mapping of object-class semantics from RGB-D video. *Journal of Real-Time Image Processing*, pages 1–11.

Viola, P. and Jones, M. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57(2):137–154.

Xiao, J., Owens, A., and Torralba, A. (2013). Sun3d: A database of big spaces reconstructed using sfm and object labels. In *2013 IEEE International Conference on Computer Vision*, pages 1625–1632.